An Incremental Interactive Algorithm for Regular Grammar Inference

Rajesh Parekh and Vasant Honavar
Artificial Intelligence Research Group
Department of Computer Science
Iowa State University, Ames, IA 50011
{parekh|honavar}@cs.iastate.edu

Introduction

Grammar inference, a problem with many applications in pattern recognition and language learning, is defined as follows: For an unknown grammar G, given a finite set of positive examples S^+ that belong to L(G), and possibly a finite set of negative examples S^-, infer a grammar G^* equivalent to G. Different restrictions on S^+ and S^- and the interaction of the learner with the teacher or the environment give rise to different variants of this task. We present an interactive incremental algorithm for inference of a *finite state automaton* (FSA) corresponding to an unknown *regular grammar.*

Search Space

A set of positive examples (strings of the unknown language) is structurally complete if each production rule of the unknown grammar is used at least once in the generation of some string in the set. If S^+ is structurally complete, it implicitly defines a lattice (ω) of candidate grammars that is guaranteed to contain the target grammar. At the base of the lattice is the *maximal canonical automaton* (MCA) that accepts exactly the set S^+. Other elements of the lattice (ordered by the *grammar covers relation*) represent progressively more general languages i.e., supersets of S^+ and are generated by successively merging states of the MCA. This (exponential sized) search space can be concisely represented by two sets $\mathcal{S}$ and $\mathcal{G}$ which correspond to the most *specific* and most *general* FSA respectively.

A *version space* based technique is used to search the hypothesis space. FSA corresponding to two lattice elements (one from $\mathcal{S}$ and the other from $\mathcal{G}$) are compared for *equivalence*. If the two FSA are not equivalent, the shortest string y belonging to the symmetric difference of their languages is posed as a *membership query* to the teacher. Based on the teacher's response to the query the learner is able to prune the search space without eliminating the desired solution. For example, if the teacher's response to a query is *negative*, the FSA accepting the negative example and all FSA that cover it (and thus also accept the same negative example) are eliminated. This elimination is carried out implicitly by modifying the two sets $\mathcal{S}$ and $\mathcal{G}$ as needed. This interaction between the teacher and learner continues till the hypothesis space is reduced to one (or a set of equivalent) FSA. The resulting FSA is provably equivalent to the target FSA.

Incremental Algorithm

The incremental version of the algorithm relaxes the *structural completeness* assumption. The teacher may provide a few positive examples to start with. The learner performs candidate elimination by posing *safe membership queries* to the teacher. After seeing more positive examples, the learner incrementally updates the lattice to incorporate the new examples and continues with candidate elimination. Eventually, when the set of positive examples provided by the teacher includes a structurally complete set for the target FSA, no more lattice updates take place and all queries are treated as safe. The algorithm then converges to the target FSA. The necessary and sufficient conditions for guaranteed convergence of the algorithm to the correct solution are identified.

Future Directions

Promising directions for further research include: heuristics for informative query generation to speed up learning; inference of regular *tree grammars* and *attributed grammars*; empirical estimates of the expected case time and space complexity of the proposed grammar inference algorithm and its extensions.

References

Parekh R.G., and Honavar V.G. An Incremental Interactive Algorithm for Regular Grammar Inference. *Computer Science Department, Technical Report TR 96-03*, Iowa State University.